REMARKS

Claims 1 - 20 are pending in the present Application. In the above-identified Office Action, the Examiner objected to the DRAWINGS as failing to comply with 37 C.F.R. 1.84(p)(5). The Examiner further rejected Claims 1, 6, 11 and 16 under 35 U.S.C. §102(e) as being anticipated by Potter. Claims 2 - 5, 7 - 10, 12 - 15 and 17 - 20 were rejected under 35 U.S.C. §103(a) as being unpatentable over Potter in view of Fesas, Jr.

The Examiner objected to the DRAWINGS because of (1) unlabeled boxes shown in Fig. 5, (2) Figs. 1 - 4 are not designated by a PRIOR ART legend and (3) reference character 432 in Fig. 4 is not mentioned in the DESCRIPTION.

The unlabeled boxes have been properly labeled. Particularly, boxes 502, 504, 506 and 508 have been labeled CPU, boxes 520, 522, 524 and 526 have been labeled BUFFER and box 540 has been labeled PHYSICAL INTERFACE. A copy of Fig. 5 as amended is attached as well as a replacement figure.

Regarding Figs. 1 - 4, Applicants submit that the objection is unwarranted since the invention is incorporated in the figures. Consequently, the figures need not be designated by a PRIOR ART legend.

Applicants reviewed both Fig. 4 and the SPECIFICATION and could not find a reference to REFERENCE NUMERAL 432. Consequently, Applicants respectfully request withdrawal of this objection.

AUS920010893US1

For the reasons stated more fully below, Applicants submit that the claims are allowable over the applied references. Hence, reconsideration, allowance and passage to issue are respectfully requested.

As stated in the SPECIFICATION, multiprocessor systems are often connected to a network or networks through a limited number (usually one) of physical interfaces. Consequently, before a processor in a multiprocessor system uses a physical interface to transmit network data, it has to first request permission to lock out all the other processors from using the interface. If more than one processor is requesting access to the interface, there may be some access contention or lock contention. To reduce the likelihood of lock contention, an algorithm is generally used to select which one of the requests to honor first. The algorithm may do so on a first-come, first serve or round robin or on a priority basis or using any other contention resolution scheme.

When an access request is honored, the requesting processor is allowed to lock out all other processors from using the interface until the data is transmitted. When the processor has finished transmitting the data, it releases the lock to allow another processor to gain access to the lock. Obviously, while the processor is transmitting data, other processors may issue requests to the lock. Hence, there may be instances when other processors have to wait before gaining access to the physical interface in order to transmit data. In these instances, the physical interface may be viewed as a

bottleneck as requests for the physical interface are accumulating at that point.

Thus, although the use of a multiprocessor in a system may greatly improve a computer system's performance, network communications performance may nonetheless not benefit from the use of the multiple processors due to this bottleneck. Therefore, it would be desirable to have a method that alleviates bottlenecks at the physical interface in the point of view of the processors. The present invention provides such a method.

According to the teachings of the invention, when a multiprocessor system that uses a limited number of physical interfaces is to transact data, a determination is made as to whether the data is network data. If the data is network data, the data is transmitted using a virtual Internet protocol (IP) address. The virtual IP address is the IP address of a data holding device rather than the address of a receiving computer.

Thus, the data before being sent onto the network to the receiving computer is sent to the data holding device. This frees up the processors of the multiprocessor system to continue to process data instead of becoming idle, waiting for the data to be transmitted. This may greatly enhance the performance of the multiprocessor system, especially in the case where there is a (long) queue to transmit data onto the network.

The invention is set forth in claims of varying scopes of which Claim 1 is illustrative.

1. A method of improving performance
in a multiprocessor system that uses a

AUS920010893US1

limited number of physical interfaces to
transact network data comprising the steps
of:

*determining whether data being
processed is network data; and*

*transacting, if the data is network
data, the data using a virtual Internet
protocol (IP) address, the virtual IP
address being an IP address given to a
data holding device in the multiprocessor
system.* (Emphasis added.)

The Examiner rejected the claims under 35 U.S.C.
§102(e) as being anticipated by Porter. Applicants
respectfully disagree.

Porter purports to teach a dynamic addressing mapping
to eliminate memory resource contention in a symmetric
multiprocessor system. According to Potter, the
granularity of a dynamic random access memory (SDRAM)
contention is typically one bank. A typical SDRAM module
has four (4) banks, each containing a fixed one-quarter of
the total memory. When a bank is accessed, it cannot be
accessed again for a certain period of time (e.g., 7 cycles
at 100 MHz). An SDRAM can support overlapping accesses to
each of its banks, where new accesses can be issued every 2
cycles, but only one access at a time per bank is possible.
To access relatively long table entries (e.g., entries
containing words), the time that a bank is tied up
increases, thereby directly increasing access contentions.
This can slow down a symmetric multiprocessor system that
is configured as a multidimensional systolic array.

A systolic array is an array of processors which are
connected to a small number of nearest neighbors in a mesh-

AUS920010893US1

like topology. For example, a one-dimensional systolic array may be regarded as a pipeline.

In a multidimensional systolic array, processors in the same position ("column") of each pipeline execute the same instructions on their input data. For a large class of applications including data networking, the processors in the same column access the same data structures. For example, a common table indicating a data communication queue must be accessible by all processors in the same column since it is not possible to know in advance which pipeline has the correct table for this input data. Therefore, access to a common memory is required among the processors of the same column.

To avoid contention and thus stalling by the processors, accesses to the common memory are scheduled. Since each processor of a column executes the same instruction code and therefore accesses the same tables in memory, the pipelines of the array are staggered (i.e., the array is configured such that a first processor of a first pipeline finishes accessing a particular memory just as a second processor of a second pipeline starts to access the same memory).

Since, however, only one access at a time per bank of an SDRAM is possible and since accesses to relatively long table entries may take up more than seven (7) cycles at 100 MHz, there will be access contentions. Potter devises a dynamic address mapping technique that eliminates contention to an SDRAM used by a symmetric multiprocessor system arranged as a multi-dimensional systolic array.

AUS920010893US1

According to the teachings of Potter, the technique defines two logical-to-physical address mapping modes that may be simultaneously provided to the processors to thereby present a single contiguous address space for accessing individual memory locations: a bank select mode and a stream mode.

The bank select mode uses high-order address bits to select a bank of a memory resource for access. A data structure, such as a table having relatively short entries, is placed within a single bank of memory and addressed using the bank select mode. Assume that the bank is "tied up" for 7 cycles during an access to a single location in the table memory. A first processor in a first pipeline of the arrayed processing engine can access a random location within this table at absolute time N. As long as the skew between pipelines is as large as the time that the bank is tied up for a single access (i.e., 7 cycles), a second processor in the same column of a second pipeline can execute the same instructions (skewed by the 7 cycles). In this case, the second processor may access the same or a different location within the table (and bank) at time N+7 without contending with the first processor.

The stream mode uses low-order address bits to select a bank within a memory resource. Here, the data structure is preferably a table having relatively long entries, each containing words that are accessed over a plurality of cycles. According to this aspect, the long entries are spread across successive banks and stream mode addressing functions to map each successive word to a different bank. By defining the table entry width as a multiple of the

access width times the number of banks, contentions can be eliminated.

For example, a processor of a first pipeline can access a first word of a random entry from a table resident in Bank 0 at absolute time N; that processor may then access a second word of the same entry from Bank 1 at time N+7. This process may continue with the processor "seeing" the entire entry as a contiguous address space. A corresponding processor of a next pipeline is skewed by 7 cycles and can execute the same instructions for accessing the same or different entry from the same table. Here, a first word is accessed from Bank 0 at time N+7, a second word is accessed from Bank 1 at time N+14, etc., without contention.

However, Potter does not teach, show or so much as suggest the steps of **determining whether data being processed is network data;** and **of transacting, if the data is network data, the data using a virtual Internet protocol (IP) address, the virtual IP address being an IP address of a data holding device** as claimed.

Fesas, Jr., the other applied reference, purports to teach a system for reducing bus overhead for communication with a network interface. According to Fesas, Jr., direct memory access (DMA) data transfers between computer systems and network interface controllers (NICs) are commonly accomplished using a technique call scatter-gather. In scatter-gather, a bus master device in the NIC is first instructed to obtain a command block from the memory of a host computer system. At a minimum, the command block contains a list of physical addresses for blocks within the

host system memory that are to be copied to the DMA device. The command block also contains a count of the number of fragments in the command block and the overall length of the data contained in the fragments pointed to by the command block. The DMA device parses the command block, extracting the address of each fragment, and transfers the fragments from the host memory to the DMA device. This process is repeated for each fragment listed in the command block until all of the data described by the command block is copied to the DMA device.

A significant performance bottleneck in using the scatter-gather technique for transferring data to a high speed network is the translation from virtual to physical addresses. Peripheral devices, such as a NIC, cannot use virtual memory addresses to effect the transfers, because the hardware to implement the virtual-to-physical address translation is typically located inside the CPU. This means that conversion between virtual and physical addresses must take place before transfers between a computer system and a NIC can take place. This conversion can take a great deal of time and consume a significant amount of the computer system's processing power. When data is passed to a device driver for transmission to the NIC, the driver first performs a virtual-to-physical address conversion for each buffer fragment passed down to it from the application layers above. It is possible for each buffer fragment to straddle physical pages of the memory system. Thus, more than one physical address may correspond to each virtual address converted. Consequently, several virtual-to-physical address

AUS920010893US1

conversions may be required for each buffer of data that is transferred from the computer system to the NIC. This can be very time-consuming because each virtual-to-physical address translation can take from tens to hundreds of CPU cycles to accomplish.

Another significant performance impediment associated with the scatter-gather technique is its command block nature. Peripheral devices such as NICs typically connect to computer systems through a peripheral interconnect bus, such as the PCI bus. In order to transfer data to or from the computer system, devices connected to the bus contend for control of the bus. Once a device is granted control of the bus, it drives bus signal lines to transfer data to or from the computer system. The performance impediment stems from the number of times a NIC must contend for the peripheral interconnect bus when transferring data using the scatter-gather technique. Under ideal circumstances for scatter-gather, bus contention to transfer data between a NIC and an attached computer system will occur three times per buffer transferred: first, when the computer system informs the NIC that a buffer is available for its use; second when the NIC reads the command block describing the buffer; and third when the NIC transfers data to or from the buffer. In typical scenarios, at least two buffer fragments will be described in each command block. As a result, there will be at least four contentions instead of three. These additional contentions create opportunities for other devices to obtain control of the bus and thus delay transfers initiated by the NIC.
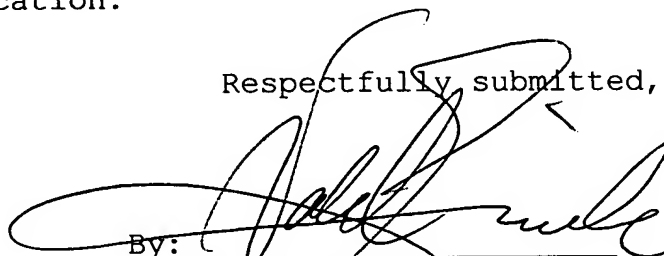
AUS920010893US1

Thus, Fesas, Jr. discloses a data transfer between a computer system and a network interface card (NIC) that occurs without virtual-to-physical address translations. According to the teachings of Fesas, Jr., the computer system allocates blocks of memory during system initialization for storing data in transit between the computer system and the NIC, and the physical addresses of these blocks of memory are stored in a table on the NIC. Consequently, address conversion is performed only once, when the memory is allocated. When a request to transfer data to the NIC is received from the upper layers, the device driver copies the data from the upper layers into the next available memory block. The device driver then formats a command and passes it to the NIC for processing. Data transfer commands are communicated to the NIC through a packet descriptor command (PDC), which is a 32-bit value subdivided into fields that completely describe the data transfer operation. The PDC contains a small ordinal value that indexes a table in the NIC, which includes a set of physical addresses of buffers preallocated by the computer system in the computer system memory. These buffers are used for storing data in transit to the NIC.

However, just as in the case of Potter, Fesas, Jr. does not teach, show or suggest the steps of **determining whether data being processed is network data;** and **of transacting, if the data is network data, the data using a virtual Internet protocol (IP) address, the virtual IP address being an address of a data holding device** as claimed.

AUS920010893US1

Since neither Potter nor Fesas, Jr. teaches the above-emboldened-italicized limitations shown in Claim 1 reproduced above, Applicants submit that Claim 1, as well as its dependent claims, should be allowable. Independent Claims 6, 11 and 16, which all incorporate the above-emboldened-italicized limitations in the above-reproduced claim 1, together with their dependent claims should also be allowable. Hence, Applicants once more respectfully request reconsideration, allowance and passage to issue of the claims in the application.

Respectfully submitted,

By: _____
Volel Emile
Attorney for Applicants
Registration No. 39,969
(512) 306-7969

AUS920010893US1

IN THE DRAWINGS:

In Fig. 5, please add the word – CPU -- above boxes 502, 504, 506 and 508. In addition, please add the word -- BUFFER -- above boxes 520, 522, 524 and 526 and the word PHYSICAL INTERFACE in box 540.

Attachment: Replacement sheet

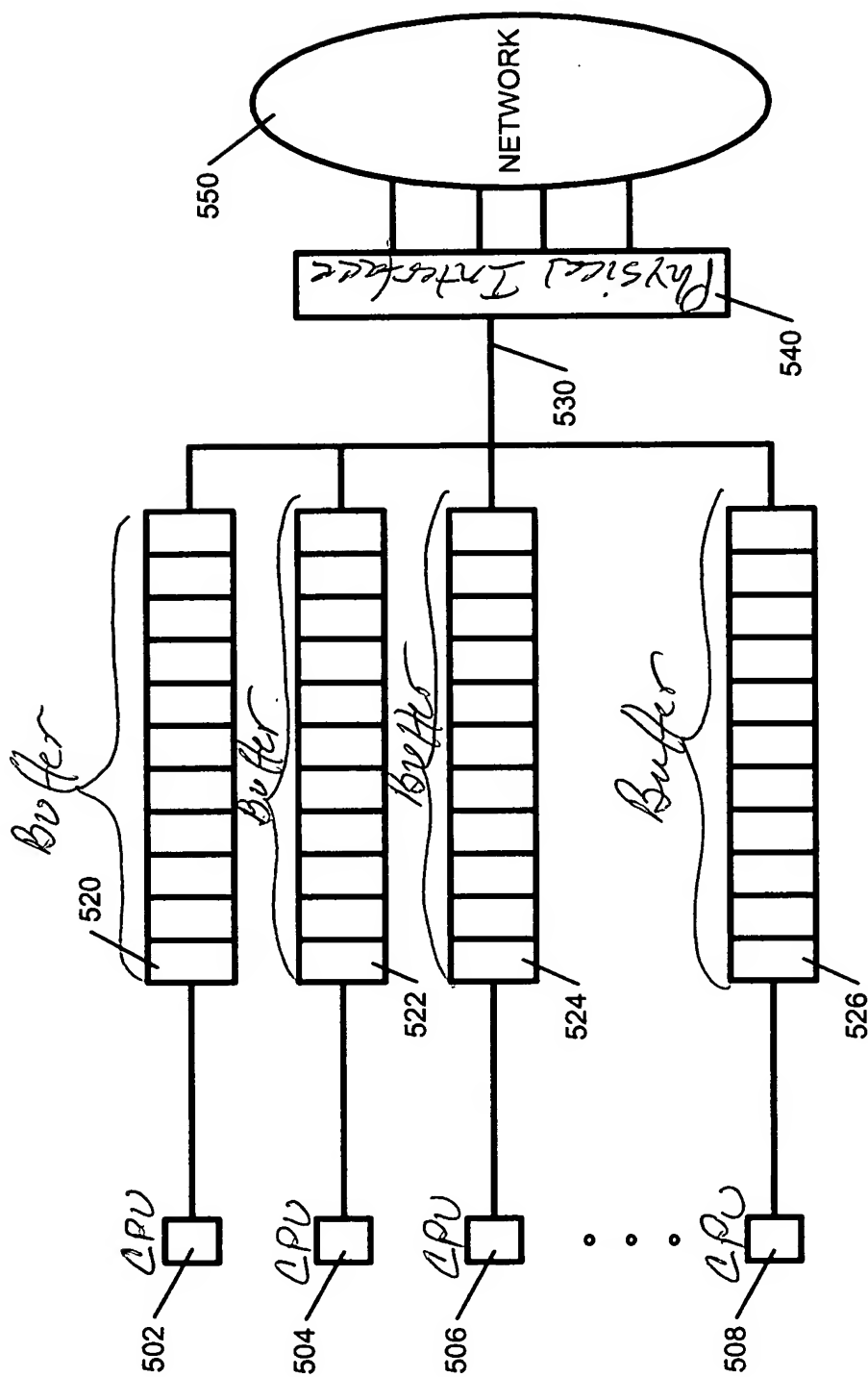Annotated sheet showing changes

AUS920010893US1

FIG. 5